AMENDMENTS TO THE CLAIMS

1.   (Original) A computer implemented system for extracting flat data from a hierarchical representation of a circuit comprising:

a processing sequence that selects an element during a traversal of a hierarchical representation of a circuit;

a processing sequence that determines if flat data for the element has been previously stored; and

a processing sequence which, if flat data has been previously stored, appends previously stored flat data to an accumulated flat data that is hierarchically and traversally related to the selected element.

2.   (Original) A computer implemented system of claim 1 wherein execution of processing sequences for selection of an element, determination of previous storage of the element and appending of stored flat data is performed recursively.

3.   (Original) A computer implemented system of claim 1 further comprising a processing sequence for masking a specified flat data stored in the accumulated flat data.

4.   (Original) A computer implemented system of claim 1 wherein the selected element's previously stored flat data describes a circuit attribute.

5.   (Original) A computer implemented system of claim 4 wherein the circuit attribute comprises a high voltage net marker.

6.   (Original) A computer implemented system of claim 4 wherein the circuit attribute comprises a super high voltage net marker.

7.   (Original) A computer implemented system of claim 4 wherein the circuit attribute comprises a conductor pad marker.

8.   (Original) A computer implemented system of claim 4 wherein the circuit attribute comprises a fuse marker.

9.   (Original) A computer implemented system of claim 4 wherein the circuit attribute comprises a negative voltage net marker.

10.   (Original) A computer implemented system of claim 4 wherein recursive selection of an element in a hierarchical traversal path terminates when no lower level hierarchical elements are found in the hierarchical path which is being traversed.

11.   (Original) A computer implemented system of claim 10

wherein the last marker found in the current hierarchical path traversal will

be the last flat data descriptor in the accumulated flat data which was

assembled during the traversal which was terminated.

12.   (Original) A computer implemented system for extracting flat

data from a hierarchical representation of a circuit comprising:

a first data structure for storing a flat data segment when an instance

of a cell is initially encountered during traversal of a hierarchical

representation;

a processing sequence for selecting a cell instance from the

hierarchical representation during traversal of the representation;

a processing sequence for determining if a selected cell instance has a

matching flat path data segment stored within the first data structure; and

a processing sequence for retrieving the flat path data segment for the

matching cell instance from the first data structure.

13.   (Currently Amended) A computer implemented system for

extracting flat data from a hierarchical representation of a circuit of claim

[[11]] 12 further comprising:

a second data structure for storing one or more sequences comprising

flat data segments which describe one or more sequence of instances from a

higher level instance in the hierarchical representation to a lower level

instance or element within a lower level instance;

a processing sequence for assembling the retrieved flat path data

segment from the first flat data structure with accumulated flat data, said

accumulated flat data comprised of flat data in the second data structure

which is hierarchically and traversally related to the selected instance thereby

revising said accumulated flat data.

14.   (Original) A computer implemented system of claim 13

further comprising

a third data structure for storing data for indicating flat data which

will not be stored in the second data structure;

a processing sequence that aborts flat data storage in the second data

structure when said sequence compares the flat data segments to be stored in

the second data structure with flat data stored in the third data structure and

finds a match between the compared flat data.

15.   (Original) A system of claim 14 wherein operations within

one or more of said processing sequences are performed recursively.

16.   (Original) A system of claim 14 wherein assembled flat data is bounded on one point by a top net and at a terminal point by a specified element.

17.   (Original) A system for generating flat data from a hierarchical representation of a circuit comprising:

a first data structure for storing a hierarchical representation of a circuit comprising instances and elements within instances;

a second data structure for storing instance flat data segments describing elements within a instance when an instance is initially encountered during traversal of said hierarchical representation;

a third data structure for storing accumulated flat data segments which describe sequences of hierarchically and traversally related elements within instances present in said hierarchical representation;

a fourth data structure for storing identifiers of an instance having a flat data segment stored in the second data structure;

a first processing sequence for selecting instances and elements of the hierarchical representation stored in the first data structure;

a second processing sequence for determining if the selected instance's flat data segment has already been stored within the second data structure by reference to the fourth data structure list of instance identifiers;

a third processing sequence for identifying and storing flat data describing the selected instance that is not listed by identifier within the fourth data structure, said third processing sequence storing the selected instance's flat data within the second data structure as well as in the third data structure, said third processing sequence adding the selected instance's flat data to sequences of hierarchically and traversally related flat data elements stored in the third data structure;

a fourth processing sequence for retrieving the flat path data segment for a selected instance which is listed in the fourth data structure from the second data structure; and

a fifth processing sequence for storing the retrieved flat path data segment and the selected instance's flat path data thereby storing revised flat path data in the third data structure.

18.   (Original)   A system of claim 17 wherein instances are comprised of cells.

19.   (Original) A system of claim 17 wherein the elements are comprised of nets.

20.   (Original) A system of claim 17 wherein the elements comprises a marker indicating a circuit attribute.

21.   (Original) A system of claim 20 wherein the circuit attribute represents a high voltage net.

22.   (Original) A system of claim 20 wherein the circuit attribute represents a super high voltage net.

23.   (Original) A system of claim 20 wherein the circuit attribute represents a conductor pad.

24.   (Original) A system of claim 20 wherein the circuit attribute represents a fuse.

25.   (Original) A system of claim 20 wherein the circuit attribute represents a negative voltage net.

26.   (Original) A system for extracting flat data from a hierarchical representation of a circuit comprising:

means for selecting elements of a hierarchical representation of a circuit and determining flat data for a selected cell element;

means for identifying repetitive cell instance elements of the hierarchical representation of a circuit and storing cell element instance identifiers in a first data structure;

means for storing the determined flat data for a selected cell element in a second data structure for storing element flat data segments for each cell element and in a third data structure for storing accumulated flat data segments which describe sequences of hierarchically related elements;

means for identifying a selected cell element with an instance identifier which matches an identifier stored in the first data structure; and

means for retrieving the flat data segment for the selected element with a matching identifier in the second data structure and appending the retrieved flat data segment with a higher level hierarchically related flat data sequence stored within the third data structure.

27.   (Original) A system of claim 26 further comprising a means for masking specified flat data being input into the second data structure.

28.   (Original) A storage medium containing program code for operating a computer to extract flat data from a hierarchical representation of a circuit by the operations comprising:

selecting an element during a traversal of a hierarchical representation of a circuit;

determining if flat data for the element has been previously stored; and

if flat data has been previously stored, then appending previously stored flat data to an accumulated flat data that is traversally related to the selected element.

29.   (Original) A method for generating flat data from a hierarchical representation of a circuit comprising:

selecting an element during a traversal of a hierarchical representation of a circuit;

determining if flat data for the element has been previously stored; and

if flat data has been previously stored, then appending previously

stored flat data to an accumulated flat data that is traversally related to the

selected element.

30.   (Original) A method of claim 29 further comprising

specifying flat data which is to be masked from input into the

accumulated flat data and storing said flat data which is to be masked into a

mask data structure;

determining if flat data for said element selected during a traversal

matches flat data in the mask data structure;

if a match is found between the mask data structure flat data and flat

data for said element during traversal, aborting the appending of previously

stored flat data to an accumulated flat data that is traversally related to the

selected element.

31.   (Original) A method for generating flat data from a

hierarchical representation of a circuit comprising:

storing a hierarchical representation of a elements of a circuit into a

data structure;

11

selecting an element from the stored hierarchical representation;

determining if the element is a cell;

if the element is a cell, storing flat path data for elements within the cell in a first flat path data structure which stores flat path segments for the cell and second flat path data structure which stores combined flat path segments;

selecting a second element from the hierarchical representation which has not been selected;

determining if the selected second element has flat data that matches flat data for an element in the first flat path data structure;

if a cell with matching flat data is found in the first flat path data structure, retrieving the flat path data segment for the cell with matching flat data from the first flat path data structure; and

storing the retrieved flat path data segment and the first flat path data structure hierarchically interconnected to the selected second element thereby storing a combined flat path data in the second flat path data structure.

32.   (Original) A method for extracting and storing reusable flat data from a hierarchical netlist comprising:

selecting a cell instance which has not been encountered during netlist traversal;

determining a flat path data describing elements along a net within the selected cell from the cell instance upper level hierarchy boundary point which is being traversed to a lower hierarchy boundary point comprised of either another cell boundary point within the selected cell or a last lower hierarchical element instance within the selected cell; and

storing the flat path data segment for the selected cell instance in a first data structure;

retrieving a stored sequence of flat data segments which describe flat data for higher level instances in the traversal sequence;

assembling the flat data segment with the retrieved sequence of flat data segments; and

storing the assembled segments in a second data structure.

33.   (Original) A method of generating flat data from a
hierarchical representation of a circuit comprising:

selecting a first element within the hierarchical representation which
has not been processed;

if the selected first element is a cell, storing an identifier for the first
element into a first data structure;

storing flat data describing flat path data comprised of element
instance identifiers for elements within the first element to a second data
structure; and

storing flat data describing flat path data comprised of element
instance identifiers within the first element within a third data structure for
storing one or more accumulated sequences of flat data segments, said each
of said sequences describes flat data for elements along a traversal path of a
net within the hierarchical representation.

34.   (Original) A method of claim 33 further comprising:

selecting second element within the representation; and

determining if the selected element has an identical identifier as an

element listed in the first data structure.

35.   (Original) A method of claim 34 further comprising:

if the first data structure contains an element identifier which matches

the selected second element's identifier, retrieving the flat data segment for

the matched element from the second data structure;

combining the flat data in the second data structure which is at a

higher level in the representation and interconnected with the higher level

segments in the third data structure with the retrieved flat data segment

thereby creating a combined flat data segment; and

replacing the flat data segment in the second data structure.

36.   (Original) A method of claim 35 wherein said sequence of flat

data segments describe one or more elements comprised of instance

identifiers for nets, cells, markers and other circuit components on a net

within the hierarchical representation which lie along a traversal path of the

net.

37.   (Original) A method of extracting flat data from a hierarchical

representation of a circuit comprising:

selecting a first element of a hierarchical representation of a circuit;

determining if the first element has the same flat data as a previously selected element, wherein said previously selected element's flat path data segment has been previously determined and stored in a first data structure;

if the first element is identical to the previously selected element, retrieving the flat path data for the previously selected element from the first data structure;

combining a copy of the retrieved flat path data segment with a cumulatively combined flat data containing a sequence of flat path data segments stored in a second data structure, wherein said combined flat data describes hierarchical elements along a hierarchical representation traversal path; and

storing said cumulatively combined flat data in the second data structure.